Semisupervised Sentiment Analysis of Tweets based on Noisy Emoticon Labels

APPROVED BY

SUPERVISING COMMITTEE:

Jason Baldridge, Supervisor

Katrin Erk

Semisupervised Sentiment Analysis of Tweets based on Noisy Emoticon Labels

by

Michael Adrian Speriosu, B.S., B.A.

REPORT

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN
August 2011

Acknowledgments

I thank Carissa Miller for partnering in an earlier closely related project. I also thank Leif Johnson for providing the 50GB of compressed tweets from which I extracted my training set. Finally, I thank Alec Go, Richa Bhayani, and Lei Huang for making public the 216 hand-labeled tweets used as the Stanford test set.

Semisupervised Sentiment Analysis of Tweets based on Noisy Emoticon Labels

Michael Adrian Speriosu, M.A.
The University of Texas at Austin, 2011

Supervisor: Jason Baldridge

There is high demand for computational tools that can automatically label tweets (Twitter messages) as having positive or negative sentiment, but great effort and expense would be required to build a large enough hand-labeled training corpus on which to apply standard machine learning techniques. Going beyond current keyword-based heuristic techniques, this paper uses emoticons (e.g. ':)' and ':(') to collect a large training set with noisy labels using little human intervention and trains a Maximum Entropy classifier on that training set. Results on two hand-labeled test corpora are compared to various baselines and a keyword-based heuristic approach, with the machine learned classifier significantly outperforming both.

Table of Contents

List of Tables

Chapter 1

Introduction

Twitter is a microblogging service where users post messages ("tweets") of no more than 140 characters. With over 175 million users generating 65 million tweets per day,[1] Twitter represents one of the largest publicly available datasets of user generated content.[2] Along with other social networking websites such as Facebook, the content on Twitter is real time: tweets about everything from a friend's birthday to a sudden earthquake can be found posted during and immediately after an event in question.

This vast stream of real time data has major implications for any entity interested in public opinion. Companies have the opportunity to examine what customers and potential customers are saying about their products and services in a naturalistic environment. Political organizations and candidates can determine what issues the public is most interested in, as well as where they stand on those issues. But hiring human analysts to comb through the staggering amount of data available on Twitter is expensive and time consuming.

[1] `http://www.pcmag.com/article2/0,2817,2371826,00.asp`

[2] Users have the option to make their posts private and readable by friends only, but many opt not to do this.

It is for these reasons that computational tools for automatically extracting relevant information from Twitter are in high demand. Existing keyword search technology sufficiently solves the problem of finding tweets about a particular topic, but automatically determining the polarity of a given tweet about that topic is much more difficult. Twitter's own search engine[3] has an option to search for tweets "with positive attitude" or "with negative attitude," but these options simply add the search term ':)' or ':(' respectively to the query. Many other simple systems exist that use lists of words like 'love' and 'hate' and assume that their inclusion indicates a particular sentiment. The best such systems may have relatively high precision, if the words they look for are truly indicative of a particular stance, but it is unreasonable to expect them to have high recall. Natural language is a complex and nuanced system, and every language has a vast and dynamic vocabulary. It is simply impossible to come up with any list of words that will capture all and only the tweets with truly positive sentiment or truly negative sentiment.

Over the past few decades, the great success of machine learning methods over symbolic, hand-built classifiers indicates that a problem such as accurately labeling tweets according to sentiment is well suited for statistical methods that learn feature weights from human-labeled instances in order to learn what features are most representative of a given class, resulting in a classifier with better coverage if not improved precision to boot. But even the task of adding sentiment tags to enough tweets to train a robust classifier in a

[3] `http://search.twitter.com`

supervised manner is a daunting task, and does not get around the problem of changing vocabulary through time and domain unless new tweets are continuously labeled. Thus, an unsupervised or semisupervised approach is ideal. This paper uses the intuition behind Twitter's emoticon-based heuristic sentiment search to assemble a large training set with noisy labels and train a Maximum Entropy classifier on that dataset. Results on a few small hand-labeled test sets are shown to outperform various baselines and a keyword-based heuristic technique.

Chapter 2

Previous Work

One of the first uses of machine learning to classify sentiment is Pang et al. (2002), in which the authors use movie review ratings as labels for the text in the reviews. They use Naive Bayes, Maximum Entropy, and Support Vector Machine classifiers to predict ratings and find the machine learning methods to outperform symbolic baselines. They also discuss some of the challenges of sentiment analysis as compared to topic detection, such as sarcasm and the "thwarted expectations" phenomenon where the reviewer's true sentiment is not revealed until the last sentence or two of the review. Pang and Lee (2008) is an updated survey of automated sentiment analysis approaches, many of which arose after the explosion of blog popularity in the past decade.

Turney (2002) classifies product reviews as either "recommended" or "not recommended." Though Turney claims an unsupervised approach, the semantic orientation of a review is calculated based on the mutual information of a review and the word 'excellent' minus the mutual information of the review with the word 'poor,' so some prior human knowledge is introduced into the classifier. Turney achieves an average of 74% accuracy on a set of reviews from Epinions.

Davidov et al. (2010) use 15 emoticons and 50 Twitter hashtags[1] to predict sentiment, confirming the accuracy of the automatic labels with human judges.

Shamma et al. (2009) obtain human labels from Amazon's Mechanical Turk service for a few thousand tweets posted during the 2008 Presidential Debates between Barack Obama and John McCain. They find that amount of Twitter activity is a good predictor of topic changes during the debate, and that the content of concurrent tweets reflects a mix of the current debate topic and Twitter users' reactions to that topic. Diakopoulos and Shamma (2010) use the same dataset to develop analysis and visualization techniques to aid journalists and others in understanding the relationship between the live debate event and the timestamped tweets.

O'Connor et al. (2010) use the subjectivity lexicon from OpinionFinder to label tweets about President Barack Obama as positive or negative and compare daily aggregate sentiment scores to the Gallup poll time series of manually gathered approval rating of Obama. Even with this heuristic labeling technique, they find significant correlation between their predicted aggregate sentiment per day and the Gallup poll.

Bollen et al. (2010) perform aggregate sentiment analysis on tweets over time, comparing predicted sentiment to time series such as the stock market and crude oil prices, as well as major events such as election day and

[1]Tokens used to indicate the topic or mood of a tweet, such as '#beer' or '#sucks.'

Thanksgiving. However, the authors use hand-built rules for classification based on the Profile of Mood States (POMS) and largely evaluate based on inspection.

Lerman et al. (2008) predict the "stock price" of political candidates in the Iowa Electronic Markets based on hand-labeled newspaper articles, improving upon baseline market prediction systems.

Efron (2004) classifies blogs according to political orientation ("left" or "right") and musical taste ("mainstream" or "alternative") based on cocitation information extracted from hyperlinks, achieving accuracy figures upwards of 90% on hand-labeled test sets for both tasks.

Bautin et al. (2008) employ machine translation techniques in analyzing the sentiment of news and blogs in many languages, based on the success of opinion mining methods for English. They find that sentiment predictions are significantly correlated across nine languages of news corpora and that differences in sentiment scores can even be used to make meaningful cross-cultural comparisons.

Chen and Lin (2010) propose methods for handling the general imbalance towards positive sentiment in most blog corpora, reflecting the observation that while most blog corpora tend to be biased towards positive sentiment, the detection of negative sentiment is of utmost importance to companies and politicians.

Hu and Liu (2004) mine large amounts of product reviews for features of that product relevant to sentiment and classify passages discussing those features as positive or negative. The effectiveness of their method is demonstrated on a corpus of product reviews.

Chapter 3

Datasets

The main hypothesis of this paper is that emoticons such as ':)' and ':(' are noisy but useful labels of sentiment. Rather than attempting to hand-label a few hundred or thousand tweets precisely, the potential for millions of tweets containing positive and negative emoticons to serve as a suitable training set is tested here. Based on the intuition behind Twitter's own advanced search "with positive/negative attitude," positive and negative emoticons are assumed to be positive and negative sentiment labels respectively, and arbitrary features are extracted from a corpus containing these emoticons. A Maximum Entropy classifier is then trained on this corpus. For evaluation, the classifier's predicted labels are compared to human supplied gold standard labels on two test sets.

3.1 Emoticon-based Training Set

The tweets used to create a training set for the experiments presented here come from a sample of the "garden hose"[1] Twitter feed, which at the time of collection streamed up to 15% of all tweets worldwide, from the period

[1] `http://dev.twitter.com/pages/streaming_api`

+	:) :D =D =) :] =] :-) :-D :-] ;) ;D ;] ;-) ;-D ;-]
−	:(=(:[=[:-(:-[:'(:'[D:

Table 3.1: The positive and negative emoticons assumed to be weak sentiment labels and used to extract tweets to create the training set.

from mid-September to late December, 2009. The lists of emoticons used as noisy sentiment labels are shown in Table 3.1. A total of 6,265,345 tweets containing at least one of these emoticons were extracted from the garden hose feed. Of these, 5,156,277 contained a positive emoticon and 1,109,068 contained a negative emoticon. Although rare, tweets with both a negative and a positive emoticon were permitted to appear twice, once for each label. To balance the training set, only 1,109,068 of the tweets with positive emoticons were kept to match the 1,109,068 tweets with negative emoticons.

Manual inspection revealed that many of the tweets in this 2.2 million tweet data set were not English, though English tweets still comprised a strong majority. As the test sets (discussed below) contain only English tweets, an attempt to eliminate non-English tweets was required. The CMU Pronouncing Dictionary[2] contains 133,354 English words including inflected forms and proper nouns. This word list was used to filter out any tweet that did not contain at least two whitespace separated tokens of length two or greater that appear on the list. While a few non-English tweets were able to pass through this filter and some English tweets with very unusual words

[2] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

or incorrect spelling were dropped, this simple check reduced the size of the training set to 1,683,161 mostly English tweets, still approximately balanced for positive and negative emoticon frequency.

After tokenizing on whitespace, unigram and bigram features were extracted from the resulting 1.7 million tweet dataset. All characters were lowercased and non-alphanumeric characters were trimmed from the left and right sides of tokens except when a token contained no alphanumeric characters, in which case it was not trimmed. To reduce noise, a standard stop list[3] of 715 very common and generally uninformative words was used to exclude such words from the unigram feature set. Bigram features were extracted before stop words were removed, however, following a hypothesis that function words such as articles, prepositions, and words like 'not' can be useful when appearing near content words. Extracting bigrams before removing stop words may also help in the capturing of certain syntactic and semantic phenomena not well captured by a "bag of words" feature set. The '$' symbol was used in bigram features to mark the beginning and end of each tweet.

Thus, the full feature set for the tweet "I love my new iPod Touch! :D" would be [*love*, *ipod*, *touch*, *$ i*, *i love*, *love my*, *my ipod*, *ipod touch*, *touch :D*, *:D $*].

[3] `http://www.ranks.nl/resources/stopwords.html`

3.2 Hand-labeled Test Sets

The first set of hand-labeled tweets used as a test set is a collection of 216 tweets on various topics collected by Go et al. (2009), a group of Computer Science graduate students at Stanford University who call their service Twitter Sentiment.[4] Of these, 33 (15.3%) labeled neutral were removed for a set of 183 tweets hand-labeled as either positive or negative. 108 tweets were positive and 75 were negative in this test set referred to as the Stanford set.

Neutral tweets were removed for two main reasons. First, while emoticons may be a good indicator of positive or negative sentiment, their absence is not a good indicator of objectivity, or lack of sentiment. It is therefore difficult to come up with a short and simple list of features that can be used to determine subjectivity versus objectivity of a given tweet, even in a noisy manner.[5] Second, subjectivity detection is largely a different problem from classification of sentiment known to be either positive or negative. The Twitter Sentiment research group regards a subjective-objective tweet classifier as future work rather than a component of their current work. In work on multi-domain sentiment classification, Blitzer et al. (2007) remove three-star reviews from a corpus of product reviews with ratings between one and five stars.

The second dataset used for evaluation comes from Shamma et al. (2009), where the authors use Amazon's Mechanical Turk service to obtain hu-

[4] `http://twittersentiment.appspot.com/`

[5] While it is possible that a hand-built lexicon of subjective words might be used to do this, this level of human intervention is restricted to baseline approaches in this paper.

man labels on 3,269 tweets posted during the Presidential Debates on September 26, 2008. Each tweet was given one or more votes from Turk users in the categories *positive*, *negative*, *mixed*, or *other*. In order to ensure relatively high inter-annotator agreement, two constraints were used to filter these tweets before their use in this paper. The first was that at least three votes must have been made for each tweet to be included. The second was that more than half of the votes must have been *positive* or *negative*; the majority label was then taken as the gold standard for that tweet. This resulted in a set of 1,898 tweets, 702 of which were positive and 1196 were negative, known as the Shamma set. Note that the Shamma set's imbalance is in the opposite direction of the Stanford test set.

In order to experiment with a test set with even greater inter-annotator agreement, another set of tweets was extracted from Shamma et al. (2009)'s original 3,269, in which all Turk users must have given the same positive or negative vote. Imposing this unanimity restriction resulted in a set of 912 tweets, 347 of which were positive and 565 of which were negative, known as the Strict Shamma test set.

Table 3.2 summarizes the statistics of each of these three test sets.

Dataset	Size	% Positive	% Negative
Stanford	183	59%	41%
Shamma	1,898	37%	63%
Strict Shamma	912	38%	62%

Table 3.2: Basic statistics about the three test sets used for evaluation in this paper.

Chapter 4

Approach

The machine learning algorithm used in this paper is Maximum Entropy (ME). The intuition behind an ME classifier is that the probability of events unseen in the training data is held uniform, i.e. that weights are estimated according to the training instances and labels while maximizing the entropy of the model. More formally, the probability $p(y|x)$ of label y given feature vector x is

$$p(y|x) = \frac{1}{Z(x)} \exp\left[\sum_{i=1}^{k} \lambda_i f_i(x, y)\right]$$

where $f_i(x, y)$ are binary feature functions (1 if x has feature i and label y is being considered, 0 otherwise), λ_i is the weight of function i, k is the number of features, and $Z(x)$ is used to normalize to a proper probability distribution.

The Maximum Entropy package from OpenNLP[1] is used to train two binary classifiers on the emoticon-based training set after the preprocessing steps mentioned above. The first classifier uses only unigram features, while the second uses both unigram and bigram features.

[1] `http://incubator.apache.org/opennlp/`

Performance of these ME classifiers is compared to two lexicon-based baselines, both using the OpinionFinder subjectivity lexicon[2] similarly to O'Connor et al. (2010). This lexicon consists of 2,304 words human judged as positive and 4,153 words human judged as negative, each considered either "strong" or "weak." The first baseline labels any tweet with more positive words than negative words as positive, and vice versa, regardless of the "strong" or "weak" status of any of these words in the OpinionFinder lists. The second baseline does the same, but only counts strong positive or strong negative words. In both cases, if the number of positive and negative words in a tweet is equal (including zero for both), the baselines back off to a random choice.

[2] http://www.cs.pitt.edu/mpqa/opinionfinderrelease/

Chapter 5

Results

Table 5.1 shows the results of the two ME classifiers, the OpinionFinder baselines, baselines that always label a tweet positive or always label a tweet negative, and the random baseline. Since there are only two classes (i.e. neutral or "mixed" tweets are ignored), the random baseline achieves 50% accuracy on average. The "all positive" and "all negative" baselines' performance reflects the label biases in the test sets from Table 3.2. It is important to note that neither of these can be considered an overall stronger baseline than the random, since the Stanford dataset has a positive bias and the Shamma datasets have a negative bias. The OpinionFinder baseline using all the words in the subjectivity lexicons outperforms the one with only strong words on all three datasets.

Both ME classifiers outperform the random and OpinionFinder baselines in all cases. The ME classifier trained on both unigram and bigram features also outperforms the all positive and all negative baselines on all datasets, as well as outperforming the ME classifier with only unigram features. On the Shamma dataset, the all negative baseline somewhat outperforms the unigrams-only ME classifier, but as mentioned above, the all negative baseline

Classifier	Stanford	Shamma	Strict Shamma
Random	.500	.500	.500
All Positive	.590	.371	.376
All Negative	.410	.629	.624
All Words OpinionFinder	.713	.588	.616
Strong Only OpinionFinder	.689	.575	.560
MaxEnt Emoticons (Unigrams)	.814	.602	.627
MaxEnt Emoticons (Uni.+Bigrams)	**.825**	**.640**	**.674**

Table 5.1: Accuracy results for baselines (random, all positive, and all negative), OpinionFinder keyword-based heuristic classifiers (all words and strong only), and Maximum Entropy classifiers trained on tweets weakly labeled with emoticons with unigrams features only and unigram and bigram features. The three test corpora are the Stanford corpus of 183 hand-labeled tweets, Shamma et al. (2009)'s corpus of 1898 tweets labeled by Amazon Mechanical Turkers, and a version of Shamma et al. (2009)'s corpus with stricter inter-annotator agreement constraints of 912 tweets.

performs unacceptably poorly on the Stanford dataset (worse than random) so cannot be considered an admissible baseline in general.

Chapter 6

Discussion

6.1 Features

Table 6.1 shows the top 20 most predictive unigram features of each class that are among the 1000 most common unigrams in the emoticon training corpus and are not themselves emoticons.[1] In other words, these are the common words most likely to co-occur with an emoticon and least likely to occur without one.

This list reveals several trends in the training corpus. The first is that non-English words such as the Spanish 'gracias' can make it through despite the filter for non-English words. Manual inspection of tweets containing 'gra-

[1]Recall that stop words have already been removed from the corpus.

+	congrats, gracias, yay, thx, smile, moon, bom, excited, awesome, hello, glad, wonderful, hehehe, loving, sweet, amazing, boa, goodnight, cute, enjoy
−	nickjonas, murphy, brittany, rip, sad, fml, triste, hurts, died, snow, headache, upset, crying, throat, poor, ugh, sucks, stomach, huhu, horrible

Table 6.1: Top 20 most predictive unigram features for the positive and negative classes that are among the 1000 most common unigrams overall and are not emoticons, in order from more predictive to less predictive.

cias' indicates that most are fully in Spanish, but the presence of words spelled the same as English words (e.g. 'me') and words borrowed from English (e.g. 'ok') allow these to pass through the filter. 'Bom' and 'boa' also appear almost exclusively in Spanish tweets. Still, the majority of the most predictive words are English.

A second observation that can be made is that not every word that is very correlated with positive or negative emoticons appears to be a general-use positive or negative word such as those often found on manually constructed lexicons like that of OpinionFinder. The first three negative words are all proper nouns ('nickjonas' usually coming from the hashtag '#nickjonas,' used to indicate that the topic of a tweet is the celebrity Nick Jonas). The high correlation with negative emoticons of 'murphy' and 'brittany' can be explained by the death of actress Brittany Murphy on December 20, 2009, a date within this dataset's timeline. Inspection reveals that the high negativity of 'nickjonas' comes from a combination of fans lamenting Jonas' not coming to a particular town, remarks about his diabetes, and tweets desperately pleading for Jonas to become a user's boyfriend.

Finally, this list shows that the nature of things users tend to associate with positive emoticons is somewhat different from those they associate with negative emoticons. Many of the positive words are general markers of positive feelings likely to be in line with native speakers' intuitions: 'yay,' 'awesome,' 'glad,' 'wonderful,' and so on. On the other hand, the negative list contains more words that describe a specific event or phenomena the user finds unpleas-

ant: 'rip' (*R.I.P.*), 'died,' 'snow,' 'headache,' 'throat,' and 'stomach.' Some more general negative words like 'sad,' 'sucks,' and 'horrible' also appear.

In addition to demonstrating the generally noisy nature of any human generated data, these trends indicate that the time period from which such data is extracted can have a strong effect on the most predictive features of the resulting classifier, and that users may express positive and negative sentiment in subtly different ways. The specific nature of such observations is unlikely to be obvious in the manual creation of a word list such as OpinionFinder's, and is another reason why the data-driven machine learning approach typically yields superior results.

6.2 Error Analysis

The OpinionFinder baselines' low coverage of various words that tend to indicate positive and negative sentiment in a given domain or corpus can be demonstrated with examples. The Stanford set contains the tweet "In montreal for a long weekend of R&R. Much needed," with a positive gold label. The only word in this tweet in the OpinionFinder lexicon is 'long,' and it is labeled a negative word. Thus, the OpinionFinder baseline incorrectly classifies the tweet as negative. Both of the Maximum Entropy classifiers trained on the emoticon dataset correctly identify this tweet as positive. While the ME classifier trained on unigrams does associate the feature 'long' with the negative class, it does so with a coefficient of only about -.005. In contrast, the feature 'weekend' is associated with the positive class with a coefficient of

.043. Similarly, the tweet "Booz Allen Hamilton has a bad ass homegrown social collaboration platform. Way cool! #ttiv" is labeled negative by the OpinionFinder baseline due to the presence of the word 'bad.' While the ME classifier trained on unigrams and bigrams assigns a negative weight to both 'bad' and 'ass,' it assigns a strong positive weight to the bigram 'bad ass' as well as both 'cool' and 'way cool.' In addition to the broad coverage of a vocabulary gleaned from millions of tweets, the fine-grained distinctions between the weights of various features, tuned on real data, give machine learning approaches an advantage unattainable by lexicon-based methods that make rough distinctions like "weak" and "strong."

The ME classifier trained on bigram features as well as unigram features achieves higher performance on both the Stanford set and the two Shamma sets. For example, the classifier trained only on unigrams incorrectly classifies the tweet "palin vs biden gonna be real good #current" as negative while that trained on both unigrams and bigrams correctly gives a positive label due to the presence of the strongly positive bigram "real good." For reasons that are not immediately obvious, the unigram-trained model assigns moderately negative weights to the unigrams 'biden' and 'current.' While the bigram model does this as well, the positivity of "real good" more than makes up for this effect.

Even the ME classifier trained on bigram features misclassifies a significantly greater portion of either of the Shamma datasets than the Stanford dataset. This, along with similarly lower performance of the OpinionFinder

baselines, suggests that there is something about the Shamma set that simply makes it more difficult than the Stanford set.

There are several reasons why this may be the case. First, both the emoticon training set and the Stanford test set are general in topic. Correct estimations of the positivity and negativity of general words in the training set like 'yay' and 'upset' are more likely to be useful in a broad-domain test set, whereas misestimations of the weights of more specific words and bigrams are likely to be washed out.

In contrast, the Shamma dataset contains a very different vocabulary distribution than the Stanford set. Unigrams like 'debate,' 'current,' and 'tweetdebate' are contained in nearly every tweet, usually in the form of hashtags such as '#tweetdebate.' Words and phrases referring to specific political issues like "health care" and "iraq war" also have a frequency orders of magnitude higher than either the emoticon training set or the Stanford test set. Thus, misestimations of the positivity or negativity of these features will be amplified in evaluation.

Further, recall that the positivity of a feature is essentially its likelihood of co-occurring with a positive emoticon like ':).' It is well known that emoticons are typically a mark of informal language, and users may be more likely to use them when discussing day to day events than when referring to political issues, further reducing the amount of data used to compute the weights of features like "health care."

Lastly, the ways in which people express their opinions about political issues may be more nuanced than the ways in which they refer to positive and negative feelings in general, simply due to the nature of political issues. Everyone agrees that a sore throat is bad, while it is less obvious how much government involvement in health care is beneficial.

Some of the reduced performance of the ME classifiers on the Shamma set as compared to the Stanford set comes from annotation errors in the Shamma set's gold labels. For example, the tweet "Retweeting @mamikaze: @Krississippi being from a military family, I do not trust Mccain's itchy trigger finger on bit. #debate08" (sic) is labeled positive by two Turkers and negative by only one. Thus it makes it into the non-strict Shamma set with a positive label, though both ME classifiers correctly label it as a negative tweet.

Chapter 7

Conclusion

This paper demonstrates a semisupervised machine learning approach to sentiment analysis of tweets that outperforms a word-list approach based on OpinionFinder, both on tweets in a general domain and on the Obama-McCain "tweet the debates" corpus from Shamma et al. (2009). A set of about 1.7 million tweets containing emoticons is used to train Maximum Entropy classifiers, where positive emoticons are noisily assumed to be positive labels and negative emoticons negative labels. A unigram and bigram feature set outperforms a unigrams-only feature set on three test sets derived from two hand-labeled corpora, and both machine learned classifiers achieve significantly higher accuracies on all three sets than the OpinionFinder baseline or more trivial baselines.

An examination of the resulting Maximum Entropy models suggests that users may refer to positive and negative feelings in different ways, with the most predictive positive words being more general and the most predictive negative words tending to reference specific negative situations. The approaches examined here perform better on the broad domain test set than on the debate test set, reflecting a partial domain mismatch between the emoti-

con training set and the debate test set and a general difficulty in classifying political language. Nevertheless, the overall hypothesis that a useful amount of predictive information is contained in the presence of a positive or negative emoticon is confirmed. It appears tractable to harness the power of large amounts of real time, user generated data in the task of extracting public opinion on political candidates, consumer products, and other entities with little to no human intervention.

Bibliography

Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. International sentiment analysis for news and blogs. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2008.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, 2007.

J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the 19th International World Wide Web Conference*, 2010.

Long-Sheng Chen and Li-Wei Lin. Two methods for classifying bloggers' sentiment. In *Proceedings of the International MultiConference on Engineers and Computer Scientists*, 2010.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

Nicholas A. Diakopoulos and David A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the 28th interna-*

tional conference on Human factors in computing systems, pages 1195–1198, 2010.

M. Efron. The liberal media and right-wing conspiracies: using cocitation information to estimate political orientation in web documents. In *Proceedings of the thirteenth ACM international conference on Information and Knowledge Management*, pages 390–398, 2004.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Unpublished manuscript. Stanford University, 2009.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

K. Lerman, A. Gilder, M. Dredze, P. A. Philadelphia, and F. Pereira. Reading the markets: Forecasting public opinion of political candidates by news analysis. In *Proceedings of the Conference on Computational Linguistics (Coling)*, 2008.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference*

on Empirical methods in natural language processing-Volume 10, pages 79–86, 2002.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

David A. Shamma, Lyndon Kennedy, and Elizabeth F. Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10, 2009.

P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, 2002.